APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/865,978	05/25/2001	Masood Mortazavi	188515/US	6345

66083   7590   02/04/2008
SUN MICROSYSTEMS, INC. c/o DORSEY & WHITNEY, LLP
370 SEVENTEENTH ST.
SUITE 4700
DENVER, CO 80202

EXAMINER
CHANKONG, DOHM

ART UNIT	PAPER NUMBER
2152	

MAIL DATE	DELIVERY MODE
02/04/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

MAILED

Application Number: 09/865,978
Filing Date: May 25, 2001
Appellant(s): MORTAZAVI ET AL.

FEB 0 4 2008

Technology Center 2100

Gregory P. Durbin
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 11/19/2007 appealing from the Office action mailed 8/22/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6804818   Codella et al   10-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form

the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2.   Claims 1, 4-11, 13-33, 35-40, 42, and 43 are rejected under 35 U.S.C. 102(e) as being

anticipated by Codella et al. (U.S. Patent Number 6,804,818), hereinafter referred to as

Codella.

3.   Some claims will be discussed together. Those claims which are essentially the same

except that they set forth the claimed invention as a computer program product, a computer

system, an apparatus, or an alternate computer-implemented method are rejected under the

same rationale applied to the described claim.

4.   Codella has disclosed:

- <Claims 1, 10, 16-18, 25, 27, 31-33, 38-40, and 43>

    A computer-implemented method for a first component to invoke a second

    component asynchronously in an object-oriented computing environment, the

    computer-implemented method comprising:

    receiving at an asynchronous proxy an asynchronous request from a first

    object-oriented component residing at a first server to invoke a second object-oriented

    component residing at a second server (column 5, lines 5-15 and column 5, line 59

    through column 6, line 7) wherein the request has a void return type and is not

    associated with application-specific exceptions (column 9, lines 28-33);

    setting an exception listener on the asynchronous proxy and a scope of the

    second component, the exception listener being registered for the second component

    (column 16, lines 31-50 and column 13, lines 55-63);

    storing the request and the scope in a queue on the asynchronous proxy

    (column 14, lines 2-6 and 13-21);

    providing a thread for identifying the received request and invoking the

    second component, wherein the thread identifies an exception listener object-oriented

    component for handling exceptions associated with the invocation of the second

    component, wherein the exception listener is registered on an asynchronous proxy, is

stateless and is operable to handle a plurality of types of exceptions from a plurality of

different components (column 14, lines 13-27 and column 16, line 63 through column 17,

line 41).

- <Claims 4 and 19>

    The computer-implemented method of claim 1, wherein the first and second

    components reside in environments allowing components to directly invoke other

    components (column 4, lines 25-51).

- <Claims 5, 14, 20, 29, 35, and 42>

    The computer-implemented method of claim 1, wherein the first and second

    components are Enterprise Java Bean components (column 4, lines 25-51).

- <Claims 6, 15, 21, 30, and 36>

    The computer-implemented method of claim 5, wherein the first and second

    components are associated with a container (column 4, lines 54-56).

- <Claims 7 and 22>

    The computer-implemented method of claim 6, further comprising placing the

    request from the first component in a queue (column 5, lines 5-15).

- <Claims 8, 23, and 37>

    The computer-implemented method of claim 7, wherein a worker thread

    dequeues the received request after receiving a transaction commit signal from the

    container (column 5, lines 59-67).

- <Claims 9 and 24>

  The computer-implemented method of claim 8, wherein the exception listener

  receives the exception and the scope of the exception (column 13, line 64 through

  column 14, line 20).

- <Claims 11 and 26>

  The computer-implemented method of claim 10, wherein the asynchronous proxy has

  the same type as the second component (column 9, lines 11-33).

- <Claims 13 and 28>

  The computer-implemented method of claim 10, wherein the first and second

  components are associated with separate servers (column 3, lines 61-67).

Since all the limitations of the invention as set forth in claims 1, 4-11, 13-33, 35-40, 42, and 43

were disclosed by Codella, claims 1, 4-11, 13-33, 35-40, 42, and 43 are rejected.

(10) Response to Argument

I.   THE REJECTION OF CLAIMS 1, 4-11, 13-33, 35-40, 42, AND 43 SHOULD BE SUSTAINED
        BECAUSE CODELLA TEACHES ALL THE ELEMENTS AS CLAIMED.

Applicant argues that Codella does not disclose three elements of the independent

claims. Specifically, Applicant argues that Codella fails to disclose an asynchronous request

having a void type return, an asynchronous request that is not associated with application-

specific exceptions, and an exception listener that is stateless. However, Applicant's

arguments should not be found persuasive for the following reasons.

A.   Codella's asynchronous request has a void-type return.

Applicant argues that Codella's teaching that "a message proxy...invokes no return type" does not correspond to the claimed feature because an asynchronous request and a message proxy are not the same. Applicant has misinterpreted Codella's teachings.

Codella discloses that the message proxy will invoke no return type when "the message bean 102 expects no result to be returned by the message proxy" [column 9 «lines 30-33»]. This teaching is quite similar to Applicant's own disclosure which states that when using void-type returns, "a client does not expect a return from asynchronous proxy 203" [see Applicant's specification, pg. 11 «lines 20-24» and pg. 12. «lines 7-8»].

Codella's message bean, like a client, is the element responsible for submitting asynchronous requests, where some of these requests will have a void type return [column 9 «lines 59-62» : a message bean defines the method with "no return type" | column 14 «lines 13-20»]. These methods correspond to the requests that are received by Codella's message proxy from the message bean [column 11 «line 66» to column 12 «line 9»].

Thus, based on the foregoing, one of ordinary skill in the art would have interpreted Codella's teaching that "a message proxy...invokes no return type" to imply that the request from the message bean also had a void-type return and was not expecting a result from the message proxy.

B.   Codella's asynchronous request is not associated with application-specific exceptions.

Applicant also argues that Codella does not teach that the asynchronous request is not associated with application-specific exceptions as claimed. Applicant's argument focuses on

Codella's teachings of an application independent field name and reiterates that a message

proxy and asynchronous requests are not the same. However, the rejection of this feature did

not rely on Codella's teaching of an application independent field.

First, a reading of Applicant's specification is necessary to better understand the

limitation that an asynchronous request "is not associated with application-specific

exceptions." Applicant's specification describes that object invocations such as an

asynchronous request do not have application specific exceptions but instead define an

"exception listener" to handle the exceptions [Specification, pg. 11 «lines 22-25»]. Traditional

methods will have defined exceptions [Specification, pg. 3 «lines 11-28»]. Thus, in light of

Applicant's specification, an asynchronous request with "no application-specific exceptions"

can be viewed as a method with no pre-defined exceptions but instead relies on an exception

listener to handle all exceptions.

Codella teaches an asynchronous request that is not associated with application-

specific exceptions because it instead relies on an associated message listener to handle all

exceptions that may result from an invocation [column 14 «lines 13-20»]. Like Applicant's

asynchronous request, Codella's request is associated with an exception listener that handles

all exceptions. Thus, Codella does an asynchronous request that is not associated with any

application specific exceptions but rather relies on a listener to handle exceptions associated

with the invocation [column 13 «lines 55-63» : the messagelistener listens for replies from the

invocation].

C.   <u>Based on Appellant's definition of stateless, Codella's exception listener is stateless.</u>

Appellant's disclosure describes that an exception listener is "stateless" is the listener can handle many types of exceptions from a variety of different components. Codella discloses the message bean listener is responsible for handling all replies, including exceptions [column 14 «lines 13-20»], from a variety of different components [column 16 «lines 31-50» | column 17 «lines 13-25»]. Codella's different types of beans correspond to components [see Appellant's claim 5]. Thus, Codella discloses an exception listener that is stateless.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

DC
1/26/08

Conferees:

/Lynne H Browne/
Lynne H Browne
Appeal Practice Specialist, TQAS
Technology Center 2100

/Bunjob Jaroenchonwanit/
Bunjob Jaroenchonwanit
Supervisory Patent Examiner
Art Unit 2152
January 31, 2008